Java - история создания, сфера применения, достоинства и недостатки

Опубликовано muff в Сб, 2011-08-20 14:01



Язык программирования JAVA

Язык программирования **Java** – это высокоуровневый объектно-ориентированный язык, разработанный в компании **Sun Microsystems**.

С момента появления языка **Java** его возможности и сферы использования значительно увеличились.

В настоящее время принято говорить о **Java** не как об отдельном языке программирования, а как о целом семействе технологий.

История создания языка Java

1991 год

Начало работы над проектом “**Green**” (Патрик Ноутон, Билл Джой, Джеймс Гослинг).

Целью проекта являлась разработка и практическая реализация компьютерной аппаратно-программной платформы нового поколения, предназначенной для персонального пользования, удобной и понятно в обращении (компьютер для обычных людей).

Основные идеи проекта:

- дружественный интерфейс;
- надежность функционирования.

Во время работы над проектом “**Green**” возникла необходимость разработать новый язык программирования, минимально ориентированный под конкретные платформы.

Новый язык программирования получил название “**Oak**” (“дуб”).

1992–1993 годы

Неудачные попытки внедрить разработанную технологию в различные сферы: интерактивное телевидение, игровые приставки и т.п.

1994 год

Создание браузера “**WebRunner**”, позволявшего исполнять небольшие интерактивные программы (апплеты), внедренные в гипертекстовые страницы

Особенности реализации:

- возможность запуска на разнообразных платформах;
- отсутствие необходимости установки приложений;
- компактный код (для сокращения времени загрузки);
- встроенная поддержка безопасности приложений

1995 год

переименование “**Oak**” в “**Java**”, “**WebRunner**” в “**HotJava**”

23 мая 1995 года

Официальное представление технологии **Java**

Реализация поддержки технологии **Java** в браузере **Netscape Navigator 2.0** (самый популярный браузер того времени)

1996 год

реализация поддержки технологии **Java** в браузере **Microsoft Internet Explorer 3.0**

1997 год

Выход существенно обновленной версии – **Java 2**.

Выделение отдельных платформ: **J2SE**, **J2EE**, **J2ME**.

Платформа включает в себя:

- язык программирования;
- набор служебных программ (JVM, JRE, JDK и т.п.);
- набор библиотек;

Java 2 Platform, Standard Edition (J2SE)

предназначается для использования на рабочих станциях и персональных компьютерах.

Является основой всех других **Java**-технологий.

Java 2 Platform, Micro Edition (J2ME)

является усеченным вариантом **J2SE**, для того, чтобы соответствовать ограниченным аппаратным возможностям мобильных устройств, таких как, например, сотовые телефоны.

Java Virtual Machine (JVM)

представляет собой программу, имитирующую работу некоторой гипотетической Java-машины.

Java Runtime Environment (JRE)

представляет собой минимальный набор средств, необходимых для запуска **Java**-программ.

Основу **JRE** составляют виртуальная машина **Java** и набор стандартных библиотек.

Java Software Development Kit (SDK)
Java Development Kit (JDK)

представляет собой набор средств, используемых в процессе разработки **Java**-программ: компилятор, отладчик, средства для автоматической генерации документации, комплект примеров и т.п.

JDK работает с уже готовыми текстами программ и не содержит какого-либо редактора.

Интегрированные среды разработки (**Eclipse**, **Netbeans**, **JBuilder** и т.п.) для компиляции программ используют внешний **JDK**

Сферы применения
Java-технологий

В настоящее время **Java**-технологии находят широкое применение в различных сферах.

Это, в первую очередь, вызвано особенностями создания и запуска **Java**-приложений.

1. Разработка приложений (application)

Язык **Java** используется для разработки обычных прикладных программ, которые запускаются пользователем и имеют доступ ко всем ресурсам компьютера наравне с любыми другими программами.

Для запуска **Java**-приложений в операционной системе должна быть установлена среда исполнения **Java (JRE)**.

2. Разработка мидлетов (midlet)

Язык **Java** используется для разработки специализированных программ, предназначенных для использования на мобильных устройствах.

Структура и возможности мидлетов обусловлены техническими особенностями мобильных устройств.

Для запуска мидлетов мобильное устройство должно поддерживать соответствующую возможность.

3. Разработка апплетов (applet)

Язык **Java** используется для разработки специализированных программ, обычно небольшого размера, запускаемых браузером внутри web-документа для вывода динамического содержимого и/или интерактивного взаимодействия с пользователем.

Для запуска апплетов браузер должен содержать соответствующий модуль (plug-in).

4. Разработка JSP-страниц (Java Server Pages)

JSP-страницы представляют собой **HTML**-документы со вставками на языке **Java**, которые используются для анализа пользовательских запросов и динамического формирования web-документов на основе результатов обработки этих запросов.

Для обработки **JSP**-страниц **Web**-сервер должен содержать соответствующий модуль.

5. Разработка сервлетов (servlet)

На языке **Java** разрабатываются специализированные программы-сервлеты, функционирующие в рамках **Web**-сервера и имеющие доступ к его ресурсам: файлам, базам данных и т.п.

Для запуска сервлетов используется **Web**-сервер со специальным модулем (контейнером сервлетов) или специальный сервер приложений.

Достоинства и недостатки
Java-технологий

Java-технологии имеют много особенностей, отличающие их от других технологий разработки программного обеспечения.

1. Переносимость

Программы, написанные на языке **Java**, после однократной трансляции в байт-код могут быть исполнены на любой платформе, для которой реализована виртуальная **Java**-машина.

Наиболее эффективно возможности реального компьютера может использовать только программа, написанная с использованием «родного» машинного кода.

2. Безопасность

Функционирование программы полностью определяется (и ограничивается) виртуальной **Java** -машиной.

Отсутствуют указатели и другие механизмы для непосредственной работы с физической памятью и прочим аппаратным обеспечением компьютера.

Дополнительные ограничения снижают возможность написания эффективно работающих **Java**-программ.

3. Надежность

В языке **Java** отсутствуют механизмы, потенциально приводящие к ошибкам: арифметика указателей, неявное преобразование типов с потерей точности и т.п.

Присутствует строгий контроль типов, обязательный контроль исключительных ситуаций.

Многие логические ошибки обнаруживаются на этапе компиляции.

Наличие дополнительных проверок снижает эффективность выполнения **Java**-программ.

4. Сборщик мусора

Освобождение памяти при работе программы осуществляется автоматически с помощью «сборщика мусора», поэтому программировать с использованием динамически распределяемой памяти проще и надежнее.

При интенсивной работе с динамически распределяемой памятью возможны ошибки из-за того, что «сборщик мусора» не успел освободить неиспользуемые области памяти.

5. Стандартные библиотеки

Многие задачи, встречающиеся при разработке программного обеспечения, уже решены в рамках стандартных библиотек. Использование объектно-ориентированного подхода

позволяет легко использовать готовые объекты в своих программах.

Для запуска приложения необходима установка **JRE**, содержащего полный набор библиотек, даже если все они не используются в приложении. Отсутствие библиотеки необходимой версии может воспрепятствовать запуску приложения.

6. Самодокументируемый код

Имеется механизм автоматического генерирования документации на основе комментариев, размещенных в тексте программ.

7. Многообразие типов приложений

На языке **Java** возможно реализовать абсолютно разные по способу функционированию и сфере использования программы.

Взято отсюда **[1]**

Источник (получено *2026-11-17 00:13*):
http://muff.kiev.ua/content/java-istoriya-sozdaniya-sfera-primeneniya-dostoinstva-i-nedostatki

Ссылки:
[1] http://coderov.net/java/75-lection-for-student/176-lection1-p1.html